FY19 Intro to Databricks Hack – Setup Guide

Introduction to using Spark on Azure Databricks

# Goals

Most challenges observed by customers in these realms are in stitching multiple services together. As such, where possible, we have tried to place key concepts in the context of a broader example.

Once all hackathon challenges are completed, you should be able to:

* Create an Azure Databricks cluster
* Create Azure Databricks workspace and notebook for data exploration
* Understand the difference between Spark 1 and Spark 2 (RDD API vs. Dataframe API)
* How to use Spark SQL API to wrangle datasets using a familiar syntax

Getting Started:

To complete the challenges, you will need the following:

• A web browser

• A Microsoft account

• A Microsoft Azure subscription

• A Windows, Linux, or Mac OS X computer

• Azure Storage Explorer

• The challenge files for this course

**Note**: To set up the required environment for the challenge, follow the instructions in the [**Setup**](https://github.com/MicrosoftLearning/databricks-intro/raw/master/Setup%20Guide.pdf) document for

this course. Specifically, you must have signed up for an Azure subscription.

Provisioning Azure Resources

**Note**: If you already have an Azure Databricks Spark cluster and an Azure blob storage account, you can

skip this section.

Provision a Databricks Workspace

1. In a web browser, navigate to http://portal.azure.com, and if prompted, sign in using the

Microsoft account that is associated with your Azure subscription.

2. In the Microsoft Azure portal, click  **Create a resource**. Then in the **Analytics** section select

**Azure Databricks** and create a new Azure Databricks workspace with the following settings:

• **Workspace name**: *Enter a unique name (and make a note of it!)*

• **Subscription:** *Select your Azure subscription*

• **Resource Group:** *Create a new resource group with a unique name (and make a note of*

*it!)*

• **Location:** *Choose any avaichallengele data center location.*

• **Pricing Tier:** Standard

3. In the Azure portal, view **Notifications** to verify that deployment has started. Then wait for the

workspace to be deployed (this can take few minutes)

Provision a Storage Account

1. In the Azure portal tab in your browser, and click  **Create a resource**.

2. In the **Storage** category, click **Storage account**.

3. Create a new storage account with the following settings:

• **Name**: *Specify a unique name (and make a note of it)*

• **Deployment model**: Resource manager

• **Account kind**: Storage (general purpose v1)

• **Location**: *Choose the same location as your Databricks workspace*

• **Replication:** Locally-redundant storage (LRD)

• **Performance:** Standard

• **Secure transfer required:** Disabled

• **Subscription:** *Choose your Azure subscription*

• **Resource group:** *Choose the existing resource group for your Databricks workspace*

• **Virtual networks:** Disabled

4. Wait for the resource to be deployed. Then view the newly deployed storage account.

5. In the blade for your storage account, click **Blobs**.

6. In the **Browse blobs** blade, click  **Container**, and create a new container with the following

settings:

• **Name**: spark

• **Public access level**: Private (no anonymous access)

7. In the **Settings** section of the blade for your blob store, click **Access keys** and note the **Storage**

**account name** and **key1** values on this blade – you will need these in the next procedure.

Create a Spark Cluster

1. In the Azure portal, browse to the Databricks workspace you created earlier, and click **Launch**

**Workspace** to open it in a new browser tab.

2. In the Azure Databricks workspace home page, under **New**, click **Cluster**.

3. In the **Create Cluster** page, create a new cluster with the following settings:

• **Cluster Mode**: Standard

• **Cluster Name**: *Enter a unique cluster name (and make a note of it)*

• **Databricks Runtime Version**: *Choose the latest available version*

• **Python Version:** 3

• **Driver Type**: Same as worker

• **Worker Type**: *Leave the default type*

• **Min Workers:** 1

• **Max Workers:** 2

• **Auto Termination:** Terminate after 60 minutes.

• **Spark Config**: Add two key-value pairs for your storage account and key like this:

fs.azure.account.key.***your\_storage\_account***.blob.core.windows.net ***your\_key1\_value***

spark.hadoop.fs.azure.account.key.***your\_storage\_account***.blob.core.windows.net

***your\_key1\_value***

**Note**: The first setting enables code using the newer Dataframe-based API to access your

storage account. The second setting is used by the older RDD-based API.

4. Wait for the cluster to be created.

Lab Files

1. Download the lab files here: [https://github.com/annedroid/DTA2018](https://github.com/annedroid/DTA2018_MLAI_databricks/blob/master/Databricks_Labs.zip)

[\_MLAI\_databricks/blob/master/Databricks\_Challenges.zip](https://github.com/annedroid/DTA2018_MLAI_databricks/blob/master/Databricks_Labs.zip)

2. Extract the Databricks\_Labs.zip file you downloaded to your local machine.

3. Ensure that the extracted folder and subfolders are NOT read-only.